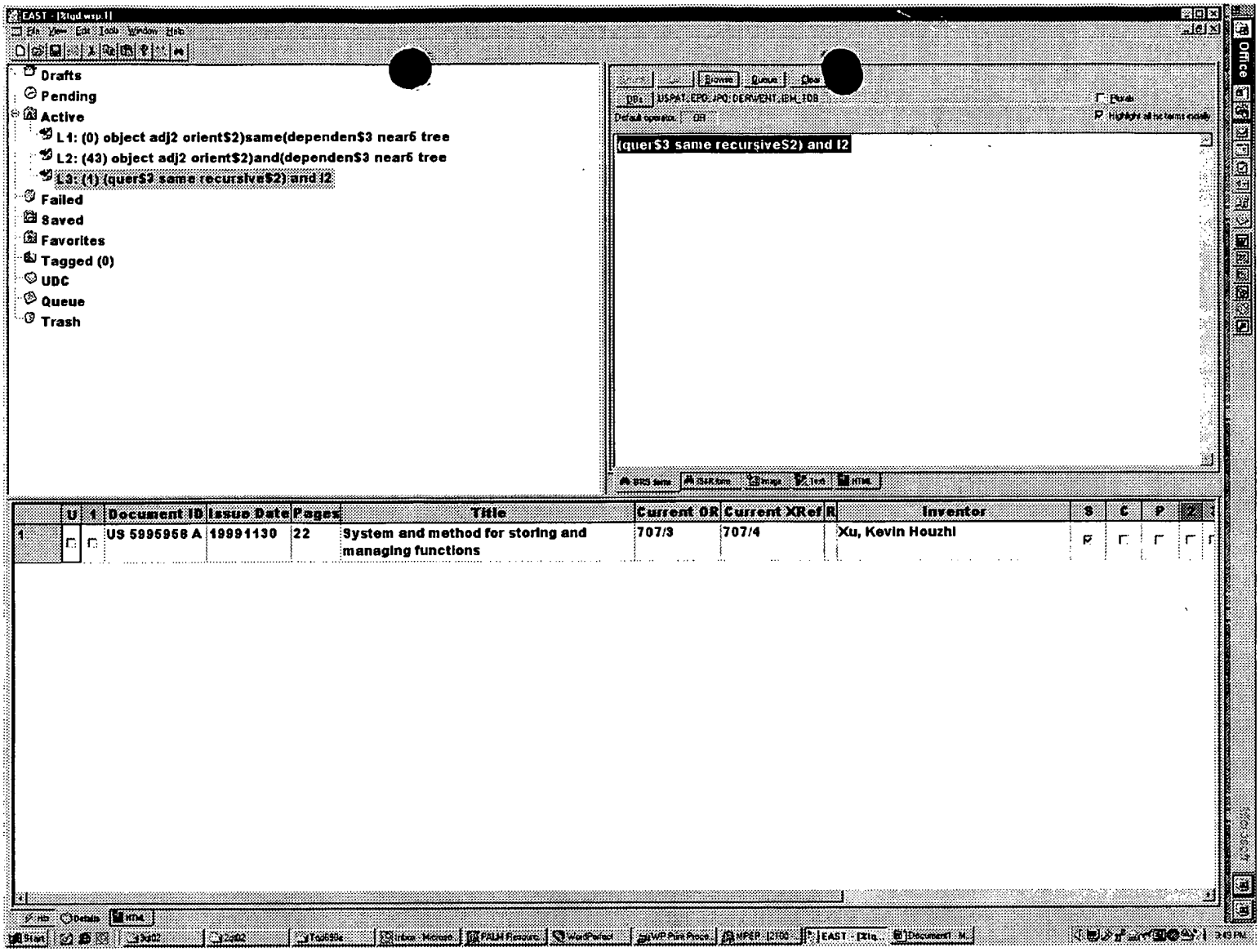
	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments	Error Definition
1	BRS	L1	1	(dependency adj tree$1) same (code! adj object$1)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/04/10 16:07		
2	BRS	L2	62	(dependency adj tree$1)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/04/10 16:11		
3	BRS	L3	7	2 and recursive$1 same path$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/04/10 16:11		
4	BRS	L5	3	4 and (object-oriented or (object! adj oriented))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/04/10 16:06		
5	BRS	L6	15	2 and (object-oriented or (object! adj oriented))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/04/10 16:06		
6	BRS	L7	2	6 and (code! adj object$1)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/04/10 16:07		
7	BRS	L4	13	2 and recursive$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/04/10 16:09		
8	BRS	L8	237	platinum.as.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/04/10 16:09		
9	BRS	L9	2	8 and recursive$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/04/10 16:09		
10	BRS	L10	0	8 and (dependency same tree$1)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/04/10 16:11		
11	BRS	L11	0	8 and (dependency and tree$1)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/04/10 16:11		
12	BRS	L12	0	8 and (dependency and tree$1)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/04/10 16:11		
13	BRS	L13	1101	recursive$1 same path$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/04/10 16:11		
14	BRS	L14	15	13 and (dependency adj (tree$1 or graph$6))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2002/04/10 16:12		


EAST - [Rigid wsp 1]
File View Edit Tools Window Help
Drafts
Pending
Active
L1: (0) object adj2 orient$2)same(dependen$3 near6 tree
L2: (43) object adj2 orient$2)and(dependen$3 near6 tree
L3: (1) (quer$3 same recursive$2) and I2
Failed
Saved
Favorites
Tagged (0)
UDC
Queue
Trash
Search
Clear
DBs
USPAT; EPO; JPO; DERWENT; IBM_TDB
Default operator:
OR
Purge
Highlight all the terms easily
(quer$3 same recursive$2) and I2
U
1
Document ID
Issue Date
Pages
Title
Current OR
Current XRef
R
Inventor
S
C
P
1
US 5995958 A
19991130
22
System and method for storing and managing functions
707/3
707/4
Xu, Kevin Houzhi
Details
HTML
Start
EAST - [Rig...
Document1 - M...
3:49 PM

BSPR:
Hierarchical and network DBMSs are no longer popular due to their limitations in database applications and due to other reasons. Relational database management systems are the most popular and effective in the modern industry.

Although relational DBMSs have many disadvantages in database applications, there hasn't had other DBMSs that could take over the leading position of relational DBMSs. Object-oriented database managements systems were introduced
under criticisms against relational database management systems. However, they
don't provide significant improvement enough in taking over the leading position.

DEPR:
If the value of a created node is a mathematical expressible function that has infinite nodes in its recursive enumeration, then the recursive enumeration is logically a part of the entire database. FIG. 8 illustrates a LDF Database 44c with a single node 250. However it illustrates infinite nodes including 251, 252, and 253, where the object with three dots 254 indicates that there are infinite number of nodes under 250. The LDF Database 44c having a single node supports infinite queries regarding the function in node 260.

DEPR:
In this section, we summarize the properties of recursive enumerations. These properties are the invariants of a database defined in Section 5, which must be protected by a database management system. These properties offer potential abilities of databases defined under lambda calculus in advanced query languages and in the implementation of a very large amount of information.

CLPV:
(ii) zero or more number of links selected from a group comprising at least a first link type and a second link type, each of said links having a tail means connecting to a node and a head means connecting to a node, each of said nodes
being attached by the head means of zero or more links of the first link type;
being attached by the tail means of zero or one link of the first link type;
being attached by the head means of zero or more links of the second link type;
and being attached by the tail means of zero or one link of the second link type, whereby the nodes and the links of the first link type forms a set of trees; and the nodes and the links of the second link type forms a set of tree, and whereby the trees reflect dependent relationships among the nodes,

US005995958A

United States Patent [19]
Xu

[11] Patent Number: 5,995,958
[45] Date of Patent: Nov. 30, 1999

[54] SYSTEM AND METHOD FOR STORING AND MANAGING FUNCTIONS

[76] Inventor: Kevin Houzhi Xu, 34 Willow Dr., Apt. 2B, Ocean, N.J. 07712

[21] Appl. No.: 08/923,849
[22] Filed: Sep. 8, 1997

Related U.S. Application Data
[60] Provisional application No. 60/030,841, Nov. 4, 1997.
[51] Int. Cl.[6] G06F 9/00
[52] U.S. Cl. 707/3; 707/4
[58] Field of Search 707/1–205

[56] References Cited

U.S. PATENT DOCUMENTS

4,558,413 12/1985 Schmidt et al. 364/973
5,430,872 7/1995 Hartle 395/200.68
5,871,537 7/1996 Rammound et al. 370/254

Primary Examiner—Thomas G. Black
Assistant Examiner—John G. Mills, III
Attorney, Agent, or Firm—Gerald J. Sarli

[57] ABSTRACT

A computer system and method for storing and managing a database that has lambda-definable functions as its foundation. The database is a collection of nodes that are connected by a set of links, and further it is a directed acyclic graph constructed from the nodes and the links. There are two types of links. By ignoring one type of the links, the database can be viewed as a set of trees; and by ignoring another type of the links, the database can be viewed as another set of trees. The directed acyclic graph and the two sets of trees illustrate the dependent relationships among the data and provide means in manipulating the database. A node has a name for the purpose of identification; and a value that carries an unit of meanings of the database application. The value of a node is selected from a regular data type, user-defined data type, and lambda-definable functions. Managing lambda-definable functions allows a database to support infinite data with finite storage. It is proved that a database is an enumeration of the properties of a lambda-definable function; and a lambda-definable function is the abstraction of a database. This gives a database management system the full computing capability that a computer has. Thus a database application doesn't need application-dependent software development to host programming languages except for high-level functional specifications, which reduces the development cost of the database application, and improves the reliability and performance of the database application.

4 Claims, 7 Drawing Sheets